

# ALGORITMOS Y ESTRUCTURAS DE DATOS

## Laboratorio 7

Se busca realizar ejercicios de conceptos asociadas a programación, manejo de estructuras no lineales e interacciones con el sistema de archivos además de la salida y entrada estándar.

### Objetivos

- Realizar los ejercicios utilizando el lenguaje de **programación C++**.
- Realizar una correcta creación y manipulación de estructuras.
- Realizar una correcta implementación de algoritmos que involucren decisiones y bucles.
- Realizar una correcta implementación de estructura de datos no lineales.

### Nomenclatura para nombre de archivos fuentes

El nombre del archivo en el cual se almacena el código fuente debe considerar el siguiente formato: laboratorioN.EXT donde; EXT es la extensión del lenguaje de programación utilizado y N el número del Laboratorio.

[Ayuda memoria](#)

---

### Ejercicio número 1

Crear una estructura para implementar un árbol binario y las funciones de crear, insertar e imprimir en orden nodos del árbol.

### Ejercicio número 2

Crear la funcionalidad que permite al usuario ingresar los números para ser incorporados en el árbol.

### Ejercicio número 3

Crear las siguientes funciones de búsqueda en el árbol: buscar valor, buscar valor mínimo y buscar valor máximo.

### Ejercicio número 4

Cree un la función de eliminar un nodo del árbol.

### Ejercicio número 5

Crear una estructura para implementar un árbol General y las funciones de crear, insertar e imprimir el árbol.

### Debe entrega además

Cree un menú para acceder a las distintas funciones desarrolladas en el presente laboratorio. Debe considerar una correcta validación para no tener errores al momento de ingresar información por parte del usuario.

## Ayuda memoria

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* left;
    Node* right;
};

Node* createNode(int data) {
    Node* newNode = new Node;
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}

void printTree(Node* root) {
    if (root == NULL) {
        return;
    }
    printTree(root->left);
    cout << root->data << " ";
    printTree(root->right);
}
```